DYNAMIC PHYSICAL LAYERS FOR WIRELESS NETWORKS USING SOFTWARE RADIO

Vanu Bose
Vanu, Inc.
Cambridge, MA
vanu@vanu.com

Roger Hu and Robert Morris
Laboratory for Computer Science
MIT
rhu@mit.edu, rtm@lcs.mit.edu

ABSTRACT

The communication parameters in mobile ad-hoc networks, such as the distance between nodes, channel characteristics, and user demands can vary quickly. Using traditional design techniques and creating a static physical layer designed to meet worst case conditions will result in poor performance and resource utilization under most operating conditions. Software radios, which implement their physical layer processing in software, provide a solution to this problem by enabling the physical layer to be modified to best meet the current conditions. This paper describes the initial work on a software radio-based ad-hoc network that allows the physical layer to be modified on a per packet basis.

1. INTRODUCTION

A software radio is a wireless communications device in which the physical and link layer functions are implemented in software. This enables a single wireless device to be reprogrammed to use different modulation, coding, and access protocols. Most software radio research to date has been driven by the interoperability problems present in commercial and military wireless systems [1] [2]. In commercial systems, the multitude of different cellular standards inhibits universal roaming and new standards are deployed slowly since it requires installing new basestation hardware and distributing new handsets to users. Similarly, the varying operational requirements of the different branches of the military require different radios and hinders the coordination of joint operations.

Software radio also provides the flexibility to adapt dynamically any aspect of the physical layer of a wireless communication system to meet the constraints imposed by the environment, network traffic load, government regulation, or user demands. Software radio is practical today in applications such as fixed customer premise equipment and wireless infrastructure where power dissipation is not a significant limitation. With predicted advances in low power processor technology, we expect that handheld software radio will be practical within five years.

In a mobile ad-hoc network, the transmission and reception requirements are constantly changing [3]. The distances between the nodes may change, physical objects may obstruct communication for a period of time, different user applications may have certain bandwidth and latency requirements, and battery life considerations can affect the data rates or distances that a node transmits.

Traditional network design, which involves designing a physical layer to meet the expected worst case conditions would result in either poor spectrum utilization, unnecessary power consumption, or inefficient bandwidth usage. Software radio solves this problem by enabling the physical layer to be changed to meet the current transmission needs.

The testbed will enable experimentation and characterization of the advantages for using software radio in ad-hoc networking. While there has been considerable research on ad-hoc networking [4], much of the work has been focused on developing efficient routing algorithms. The goal of this testbed is to start with a fairly simple, but extensible, software radio system for ad-hoc networking and evolve the system based on the analysis of actual usage patterns.

This system enables the modulation and channel coding used for transmission of the payload to be varied on a per packet basis. Changing the modulation on every packet may not be the most efficient approach, but it does permit considerable experimental flexibility. The flexibility of the initial testbed is limited to modulation and channel coding, though it can be expanded to incorporate other functions such as multiple access technique, encryption and source coding in the future. The header for each packet in the system utilizes a fixed, known modulation. The header contains information describing the modulation and coding used for the payload. A more sophisticated system might eliminate the overhead of transmitting the coding and modulation in the header of each packet, and only signal a change when it is needed. Some initial work on a protocol to support such dynamic changes can be found in [5].

2. FRAMEWORK

This section presents a framework for representing the constraints imposed on a communications system and describing how they affect the choice of coding and modulation. The constraints come from a number of sources including regulation, channel conditions, and user demands. Regulation may impose bandwidth and transmitted power limits, while the channel background noise power imposes limits on achievable data rates. User constraints cover a wide range of issues including required data rate, tolerable latency, acceptable error rate and desired battery life. Tables 1, 2 and 3 summarizes these constraints.

Constraint	Type	Variable
Bandwidth	max	BW
TX Power	max	P_{tx}

Table 1. Constraints imposed by regulation.

Constraint	Type	Variable
Latency	max	τ_L
Data Rate	min	R
Error Rate	max	p_e
Power Dissipation	max	P_D

Table 2. Constraints imposed by the user.

Constraint	Variable
Noise Power	N_0
Transmit Distance	d_{tx}

Table 3. Constraints imposed by the channel.

These constraints are used to determine the physical layer functionality. We assume that the actual values for the constraints are set by a higher level function. This framework describes how a link is utilized for a particular transmission and should not be confused with a quality of service (QoS) metric, which describes how the network is shared.

We assume that the distance to the node is the only piece of information known about the receiving node and can be estimated by using the received signal strength of the last transmission from that node. Because state management in an ad-hoc network can impose a significant overhead, we assume little knowledge about the rest of the network is needed. The goal is to take traffic traces from this testbed and determine what set of information about the rest of the network is needed to improve performance.

Next, we explore how how each physical layer function impacts these constraints. For simplicity, we have limited the functions to two components: channel coding and modulation. Each combination of physical layer functions produces values for each of the constraint variables.

A given modulation format has a modulation efficiency ($bits_{sec \cdot Hz}$), probability of error that can be expressed as a function (f_{err}) of transmitted power (P_{tx}), noise (N_0), and distance to the receiver (d_{tx}) which is used to estimate the signal strength at the receiving node. In addition, the modulation function incurs a latency (τ_{mod}) and consumes a certain amount of power (P_{mod}) for each bit processed. The parameters for the modulation function are listed in table 4.

Constraint	Variable
Bandwidth Efficiency	$b_{sec \cdot Hz}$
Probability of error	$f_{err}(P_{tx}, N_0, d_{tx})$
Latency per bit	τ_{mod}
Power consumption per bit	P_{mod}

Table 4. Parameters associated with a modulation function.

The parameters for the channel coding operation are presented in table 5. The coding rate (R_c) parameter is the number of input bits divided by the number of output bits. For a given code, we can calculate the probability of a detected error p_d and the probability of an undetected error p_u [6]. Coding can be a computationally intensive operation, and the power consumption (P_{coding}) and latency (τ_{coding}) per bit are important considerations.

Constraint	Variable
Code rate	R_c
Pr of detected error	p_d
Pr of undetected error	p_u
Power consumption per bit	P_{coding}
Latency per bit	τ_{coding}

Table 5. Parameters associated with a coding function.

The next task is to map the parameters for a given choice of modulation, coding, channel noise, allowed bandwidth and transmit power to the user specified constraint set of latency, data rate, power dissipation and tolerable error rate. The latency can be calculated by summing the computational latencies from the modulation and coding with the actual time required to transmit the signal, determined by dividing the distance by the speed of light, c. The user specified latency τ_L, is an upper bound, since it represents the maximum tolerable latency.

$$\tau_L \geq \tau_{mod} + \tau_{coding} + \frac{d_{tx}}{c} \tag{1}$$

The power dissipation (P_D) has two components: the transmit power and the computational power. The transmit power is the sum of the power required to transmit the header and the payload, and the computational power is the sum of the power required to compute the coding (P_{coding}) and modulation (P_{mod}) for the payload, which are expressed as power per bit transmitted. We have assumed that the power required to compute the header is negligible and there is a fixed length and modulation type for the header.

$$P_D \geq P_{tx} * N_{header} + (P_{tx} + P_{coding} + P_{mod}) \cdot N_{payload} \tag{2}$$

We have made the simplifying assumption that packets are sent and received in the same format. Thus, the computational power is an average of the power required for transmitting and receiving.

Data rate refers to the user data rate, not the channel rate, so we have to account for the reduction in user data rate due to channel errors and packet framing. The actual user data rate is the raw channel rate, multiplied by the fraction of the transmission that comprises the payload, the coding rate, the fraction of the bits that are expected to be received correctly and the expected fraction of the bits that will not have to be retransmitted. We have assumed that a simple ARQ scheme is used without forward error correction. The user specified rate, R, is a lower bound on the data rate.

$$R \leq bits_{sec \cdot Hz} \cdot BW \cdot \frac{N_{payload}}{N_{packet}} \cdot R_c \cdot$$

$$(1 - f_{err}(P_{tx}, N_0, d_{tx}) \cdot p_d) \tag{3}$$

The final important constraint is the probability of error, which is simply the probability of error for the choice of modulation multiplied by the probability that the error is undetected. The user specified probability of error constraint is an upper bound on the allowable probability of error.

$$p_e \geq f_{err}(P_{tx}, N_0, d_{tx}) \cdot p_u \tag{4}$$

The four simultaneous inequalities for latency, power dissipation, rate and probability of error define a region in this four dimensional space that meets the users requirements. A suitable pair of coding and modulation functions can be found by mapping the possible combinations into this space and choosing one that falls within the acceptable region.

If the set of allowable modulation and coding standards is small, the search for an appropriate combination can be fairly simple. However, as low power processors continue to improve, it will be practical to permit a much wider range of functions and combinations, which will require a more sophisticated search algorithm. Note that it is not practical to pre-compute these combinations, as they depend upon dynamic factors such as channel noise level and distance between nodes that can take on a wide range of values. This framework can also be applied to systems based on ASICs that have a few coding or modulation types built in. In this case, the search space would be small, and the appropriate combination can be selected from the built-in functions.

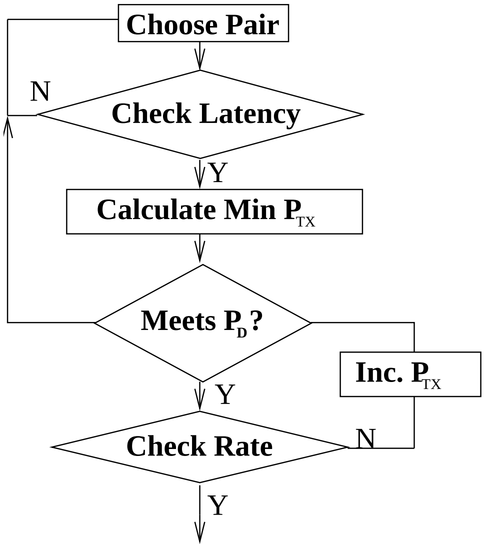

Fig. 1. Example design process.

3. EXAMPLE

This section presents an example of how the framework can be used to select different modulation and coding pairs for different packets. We have chosen a simple iterative design process (illustrated in Figure 1) that will verify if a particular choice of modulation and coding meets the constraints imposed by regulation, the user, and the channel. There are three variables to manage in the design process: coding, modulation, and transmitted power.

The first step is to choose a coding/modulation pair with bandwidth efficiency and coding rate that could support the required data rate. We then use equation 1 to see if the pair meets the user imposed latency constraint. If the inequality is not satisfied, then we choose another pair and start over the process.

If the latency inequality is met, the next step is to determine the transmit power required. Because all of the parameters in inequality 4 are known except for P_{tx}, we can determine the minimum transmit power required to meet the probability of error constraint by solving for P_{tx}.

The value calculated for P_{tx} can be used to check that the power dissipation constraint of inequality 2 is met. If not, we must return to the first step and try another pair. Otherwise, we check the rate constraint described in inequality 3. If the inequality is not met, we can try incrementing the transmit power to help reduce the number of errors. We must then re-check that the power dissipation constraint (inequality 4) still holds.

Tables 6 and 7 present measurements of power consumption and latency for several different modulation functions on a Pentium III/733 MHz. This processor is not the best choice for many software radio applications, but the results provide us with an initial set of benchmarks for comparison. Cycles counts can be used to estimate the power dissipation, since these counts can be converted to power by using an estimate of joules per cycle for a particular processor. The StrongARM, for instance, can be as low as 1 nJ per instruction, though a Pentium processor is considerably higher.

Type	Cycles	Latency (ns)
BPSK	891	1244.8
DQPSK	1669	2277.33
4-QAM	962	1353.6
8-QAM	1063	1411.2
16-QAM	1202	4910.0

Table 6. Cycles per sample and latency for several modulation functions.

Type	Cycles	Latency (ns)
(23, 12,7) Golay	6877	9381.90
(24, 12, 8) Golay	7365	10047.02
(18, 6, 8) Golay	2964	4043.43
(15, 11, 3) Hamming	4151	5663.37
(10, 6, 3) Hamming	1695	2311.90
(16, 8, 5) cyclic	3166	4318.96
(207, 187) Reed Solomon	177662	242376.9

Table 7. Cycles per sample and latency for several encoding functions.

Practically, it will not be necessary to re-calculate the coding and modulation pair for each packet. The appropriate pair could be determined for each flow through the node and then updated only when the channel constraints change appreciably.

4. SUMMARY

A flexible physical layer for an ad-hoc network provides the ability to vary the modulation and coding dynamically, providing potential improvements in performance, more efficient bandwidth, and extended battery life.

The framework presented in this paper is fairly rudimentary. There are many other functions that should be included, such as forward error correction and different access protocols. However, the goal of this framework is to define a fairly simple system that can be easily built and tested. Data from the testbed will be used to drive the further development of the constraint framework, dynamic design process, and the network state management. The knowledge of the battery life at the receiving node will be an important consideration, and the degree to which this state must be maintained throughout the network is an important characteristic that must be determined. The goal is to mature the system to the point that it can be used to perform experiments that help determine how the flexibility of software radios might influence the performance of various access protocols and routing algorithms in ad-hoc networks.

5. ACKNOWLEDGEMENTS

The authors would like to thank Andrew Chiu and Alok Shah for their assistance in preparing and revising the paper. This material is based upon work supported by the Defense Advanced Research Projects Agency under Contract No. F30602-99-C-0173 and a grant from the Nippon Telegraph and Telephone Corporation.

6. REFERENCES

[1] Joe Mitola, "Challenges in the Globalization of the Software Radio," *IEEE Communications Magazine*, 1st qtr. 1999, to appear.

[2] Raymond J. Lackey and Donal W. Upmal, "Speakeasy: The Military Software Radio," *IEEE Communications Magazine*, vol. 33, no. 5, pp. 56–61, May 1995.

[3] Scott Corson, "Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," January 1999.

[4] IETF Secretariat, "Mobile Ad Hoc Networking (MANET) Charter," January 2001.

[5] Andrew G. Chiu, "Adaptive Channels for Wireless Networks," M.S. thesis, Massachusetts Institue of Technology, Cambridge, MA, June 1999.

[6] Stephen B. Wicker, *Error Control Systems*, Prentice Hall, 1995.